

**Goals** Isabelle's classical reasoner employs introduction, destruction, and elimination rules to perform proofs. You learn how these rules work by applying them step-by-step. Moreover, proofs on recursive functions by induction and case analysis are recapitulated.

## Exercise 1 [4] Proofs with introduction and elimination rules

In the exercise theory, there is a number of lemmas of propositional and predicate logic. Prove these lemmas with single-step applications of introduction and elimination rules.

## Exercise 2 [4] Datatypes, Recursive functions, Induction, and Case Analysis

We define our own copy of the list datatype

```
datatype 'a mylist = Mynil | Mycons 'a "'a mylist"
```

Define functions `myapp` and `myrev` to append and reverse lists. Prove the lemma

```
myapp xs ys = Mynil  $\leftrightarrow$  xs = Mynil  $\wedge$  ys = Mynil
```

and the following associativity lemma:

```
myrev (myapp xs ys) = myapp (myrev ys) (myrev xs)
```

To prove the latter, you will probably need to prove two auxiliary lemmas. If we reduce the goal as much as possible and look at the goal state, we see that one of the auxiliary lemmas should be `myapp xs Mynil = xs`. The second auxiliary lemma can be found the same way.

## Exercise 3 [2] Datatypes, Recursive functions, Induction, and Case Analysis

For our list datatype from the previous exercise, we defined functions `mylen`, `myset`, and `mynth` to compute the length, the set of all elements, and the  $n$ -th element of a list.

Prove that the  $n$ -th element of a list is in the set of all elements of a list. What conditions on  $n$  need to be fulfilled for this lemma to hold?